

IN THE CLAIMS:

1. An object-oriented computing system, comprising:

a data file structure stored in a memory which contains an identification of objects in a software program, properties and handlers associated with respective objects, pointers to ancestor objects from which a given object inherits properties and handlers, and pointers to descendant objects which inherit properties and handlers from a given object;

a user interface via which a user selects an object and which displays ancestor objects and descendant objects for the selected object;

means associated with said interface for enabling a user to indicate a change in relationship between a selected object and at least one ancestor or descendant object; and

means responsive to an indicated change for modifying one or more pointers associated with the selected object in said data file structure to correspond to said change.

2. The object-oriented system of claim 1 wherein said indicated change is the addition of a new object as an ancestor object, and wherein said modifying means is responsive to said indicated change to add a new object to said data file structure and to add a pointer to the selected object which points to said new object.

3. The object-oriented system of claim 1 wherein said indicated change is the splicing of an object between the selected object and an ancestor object, and wherein said modifying means is responsive to said indicated change to change a pointer associated with the selected object from one which points to the ancestor object to one which points to the spliced object.

4. The object-oriented system of claim 3 wherein said modifying means further changes a pointer associated with said ancestor object from one which points to the selected object to one which points to the spliced object.

5. The object-oriented system of claim 1 wherein said user interface further enables a user to move selected properties and handlers from one object to another, and said modifying means is responsive to such a move to change the association of properties and handlers in said data file structure from said one object to said other object in a corresponding manner.

6. The object-oriented system of claim 1 wherein said data file structure comprises a database in which each object forms the basis for a record in the database, and each record contains fields which respectively identify the pointers, handlers and direct ancestor objects of the object with which the record is associated.

7. The object-oriented system of claim 6 wherein each record also contains a field that identifies the direct descendant objects of the object with which the record is associated.

8. The object-oriented system of claim 1 wherein said user interface comprises:

a window that is displayed on a display device of the computer system, said window including:

- a first panel which identifies an object of interest,
- a second panel which identifies objects that are parent objects for the object of interest, and
- a third panel which identifies objects that are children objects for the object of interest.

9. A user interface for designing the structure of objects in an object-oriented computer system, comprising:

    a window that is displayed on a display device of the computer system, said window including:

        a first panel which identifies an object of interest,

        a second panel which identifies objects that are parent objects for the object of interest, and

        a third panel which identifies objects that are children objects for the object of interest;

    means for enabling a user to add or delete an object identified in one or more of said panels; and

    means responsive to a user action indicating the addition or deletion of an object to one of said panels for modifying the structure of said object-oriented computer system in a corresponding manner.

10. The user interface of claim 9 wherein said enabling means includes a cursor on said display device that is controlled by a user-activated input device.

11. The user interface of claim 9 wherein said enabling means further enables a user to splice an object between the object of interest and its identified parent objects, and said modifying means is responsive to a splicing command from a user to add a new level to the structure of the object-oriented system and insert an object at the new level.

12. The user interface of claim 9 wherein said enabling means includes a menu of commands in said window via which a user controls said enabling means to add or delete a parent or child object for an object of interest identified in said first panel.

13. The user interface of claim 9 wherein said enabling means and said modifying means includes a drag-and-drop function via which a user can move identification of objects into, within and between any of said panels to modify relationships between objects.

14. The user interface of claim 13 wherein said second panel identifies at least two objects that are parents of an object of interest, and wherein said drag-and-drop function enables a user to change a relative order of precedence among said parent objects.

15. A method for redesigning the structure of an object-oriented program in a computer, comprising the steps of:

storing in a memory a data file structure which identifies objects in said program, properties and handlers respectively associated with said objects, and parent objects from which a given object inherits properties and handlers;

displaying a user interface which identifies an object of interest, parent objects for said object of interest, and children objects which inherit properties and handlers from said object of interest;

indicating via said user interface a change in relationship between said object of interest and at least one parent or child object for the object of interest; and

modifying said data file structure in response to said indication to identify said change in relationship.

16. The method of claim 16 wherein said change in relationship is the addition of a new level to said structure, and wherein said modifying step includes the steps of adding an object to said data file structure, identifying at least one parent of the object of interest as a parent of the added object, and identifying the object of interest as a child of the added object.

17. The method of claim 15 wherein said indicating step comprises the steps of dragging an indication of an object from one location on said user interface to a different location on said interface which is associated with a different relationship relative to the object of interest, and dropping the dragged indication at the different location to effect the change in relationship.

18. The method of claim 17 wherein said interface identifies at least two parent objects for the object of interest, and said dragging and dropping steps comprise changing a relative order of precedence for said parent objects.